

**APPLICATION**  
**FOR**  
**UNITED STATES LETTERS PATENT**

**TITLE:           METHOD AND APPARATUS FOR EFFICIENT  
CONVERSION OF SIGNALS USING LOOK-UP TABLE**

**INVENTOR:     JAEHOON HEO**

Express Mail No. ER439208838US

Date: March 8, 2004

METHOD AND APPARATUS FOR EFFICIENT CONVERSION OF SIGNALS USING  
LOOK-UP TABLE

[0001] This application claims priority under 35 USC § 119(e)(1) of Provisional  
5 Application No. 60/452,397, filed March 6, 2003.

Field of the Invention

[0002] The present invention generally relates to conversion of information signals. More  
10 particularly, the invention relates to the efficient conversion of audio signals using a look-up  
table. Still more particularly, the invention relates to the use of a look-up table to efficiently and  
quickly convert direct stream digital (DSD) signals to pulse code modulated (PCM) signals.

Background of the Invention

15 [0003] Audio systems capable of reproducing digital format signals allow high fidelity  
sound and theater like effects compared to audio systems that can reproduce analog format  
signals. New digital audio disk formats such as super audio compact disk (SACD) allow  
reproduction of an extended range of frequencies compared to that of the more conventional  
20 digital compact disk (CD). The SACD contains a DSD signal while CDs contain PCM signals.  
Each sample of an analog audio signal is translated into either of the two binary digits 0 or 1 in a  
DSD signal encoder. The number of 0 and 1 binary digits over a given period of time determines  
the value of the analog audio signal.

[0004] PCM audio signal samples on a CD are translated into 16 bits representing the  
25 value of the analog audio signal. PCM signals have a uniform time period between samples,  
with the rate of sampling varying from 4 KiloHertz (KHz) to 192 KHz. DSD signals are sampled  
at much higher rates such as 2.8224 MegaHertz (MHz). Thus, an audio signal sampled using a  
PCM system at 44.1 KHz sampling rate and a DSD system at 2.8224 MHz sampling rate would  
result in 64 DSD samples occurring between each PCM sample.

30 [0005] Sampling an analog audio signal into a DSD signal may be performed by a sigma-  
delta modulator. A sigma-delta modulator includes analog circuitry that captures the analog  
audio signal and converts it into a single bit stream. Because of its single bit format, DSD  
signals can be converted back into the analog audio signal using minimal hardware. However,

manipulation of the single bit DSD stream representing the analog audio signal can be very difficult. For example, tasks such as increasing the volume, adjusting treble or bass, etc. is very difficult because the DSD signal cannot be easily processed using existing digital filters and digital signal processing (DSP) techniques. One solution is to convert the DSD signal into a PCM signal using a Finite Impulse Response (FIR) digital filter. PCM signals can be processed using digital filters and DSP techniques to allow manipulation of the PCM signal to accomplish tasks such as increasing the volume or adjusting bass and for more complex tasks such as surround sound effects. After manipulation of the PCM signal, the signal may be converted back to a DSD signal and/or into analog audio signal format and transmit to speakers.

**[0006]** Conversion of the DSD signal to a PCM signal may require a high quality and expensive FIR digital filter containing large quantities of complex hardware. The DSD to PCM converter may have an odd sized binary multiplier for multiplying 1 bit by the number of bits needed to encode the PCM signal (i.e. 1 by 16 bit multiplier, 1 by 24 bit multiplier, 1 by 32 bit multiplier, and so on...). The DSD to PCM converter may also include a sign controller and an N-coefficient buffer to implement the FIR filter.

**[0007]** Digital signal processors that do not contain the dedicated hardware described above for DSD to PCM conversion are not capable of efficiently performing this conversion. Thus, there has been a longfelt need for an improved and low-cost method implemented in software or firmware and apparatus for efficient conversion of DSD signals to PCM signals in a digital signal processor (DSP).

### Summary of the Invention

**[0008]** The problems noted above are solved in large part by a method for conversion of signals that includes receiving a first plurality of bits from a first signal. The first signal may be a direct stream digital (DSD) signal. The method also includes performing a look-up in a table with a first subset of bits in the first plurality of bits to generate a result and adding the result to a sum. The subset of bits is a word. The method further includes performing another look-up in the table with the next subset of bits in the first plurality of bits and adding the result to the sum until a look-up with a last subset of bits in the first plurality of bits is performed and the result added to sum. Next, the method comprises providing the sum as a first multiple bit value of a

second signal. The second signal may be a pulse code modulated (PCM) signal. The method also includes receiving a second plurality of bits from the first signal and converting to a second multiple bit value of the second signal using the steps described above until all bits in the first signal have been converted.

5   **[0009]**        In the preferred embodiment of the invention, the table is a two dimensional array containing a plurality of elements. Preferably, the size of the first dimension equals the number of bits in the plurality of bits divided by the number of bits in the subset of the plurality of bits and the size of the second dimension is equal to  $2^{(\text{number of bits in subset})}$ . Each element in the table contains one multiple bit result. Preferably, performing the look-up in the table comprises  
10   accessing the element in the array that corresponds to the number of the subset in the plurality of bits and the value of the subset of bits.

**[0010]**        An apparatus for conversion of signals is described that includes a first-in-first-out (FIFO) buffer that contains a plurality of bits from a first signal, the plurality of bits further divided into a plurality of subset of bits of the same size. Each subset of bits is a word. The first  
15   signal may be a DSD signal. The apparatus comprises a look-up table coupled to the FIFO buffer, the look-up table generating a result for each of the plurality of subset of bits. In the preferred embodiment of the invention, the apparatus includes an accumulator coupled to the look-up table, the accumulator holding the results added together. After adding the result for the last subset of bits in the plurality of bits, the accumulator generates at an output a multiple bit  
20   second signal. The second signal may be a PCM signal.

**[0011]**        The apparatus for conversion of signals also includes an address generator connected to the FIFO buffer and look-up table. Preferably, the address generator provides to the look-up table the address of a section in the look-up table corresponding to each of the plurality of subset of bits. Each section includes a plurality of results for each subset of bits, with one of  
25   the plurality of results selected by the value of the subset of bits. The address of each section in the look-up table corresponding to each of the plurality of subset of bits is sequential. In some embodiments of the present invention, the look-up table is contained in a memory located on a DSP. In alternative embodiments of the invention, the look-up table is contained in an external memory coupled to the DSP.

[0012] An object of the present invention is to provide a method and apparatus for efficiently converting DSD signals to PCM signals using a look-up table.

[0013] The present invention provides significant advantages over the prior art. One advantage is the simplified circuitry and elimination of special hardware (e.g. 1 bit by 32 bit multiplier) for conversion of DSD signals to PCM signals. Another advantage is the reduced processor resources and bandwidth needed to convert signals using the present invention. The apparatus and method of the present invention can process multiple DSD sample bits (e.g. 16 bits, 32 bits, 64 bits, and so on) to a PCM signal sample in one clock cycle resulting in much faster conversion. Finally, because the present invention may be implemented in firmware or software, another advantage is that modifications to the firmware or software code can be easily and quickly performed.

#### Brief Description of the Drawings

[0014] The preferred embodiments of the invention will now be described with reference to the accompanying drawings in which:

Figure 1 is a block diagram of a DSD to PCM conversion device containing an N-coefficient buffer;

Figure 2 is a block diagram of a DSD to PCM conversion device that uses a look-up table in accordance with the preferred embodiment of the invention;

Figure 3 shows a flow chart for generating the look-up table shown in Figure 2 in accordance with the preferred embodiment of the invention; and

Figure 4 shows a flow chart for conversion of DSD signals to PCM signals using the look-up table generated in Figure 3 in accordance with the preferred embodiment of the invention.

#### Detailed Description of Preferred Embodiments

[0015] Figure 1 is a block diagram of a DSD to PCM signal conversion device 100. The device includes an N-bit delay line 110 that may be implemented as a bit addressable register. N sequential DSD bit samples are stored into the N-bit delay line 110 for conversion to a PCM signal sample 180. DSD signal bit 105 is the most recently sampled DSD bit. The N-bit delay

line **110** receives a plurality of DSD signal bits that may have been sampled from the analog audio signal at a rate of 2.8224 MHz.  $N$ -bit delay line **110** may be capable of holding  $N=32$  bits. Alternatively, the  $N$ -bit delay line **110** may hold  $N=16$  bits,  $N=64$  bits,  $N=128$  bits, and so on.

[0016] DSD to PCM signal conversion device **100** also includes an  $N$ -coefficient buffer **120**. The  $N$ -coefficient buffer includes coefficients  $c1$  **125a**,  $c2$  **125b**,  $c3$  **125c**, ..., and  $cN$  **125N** that are each 32 bits in length **124**. Coefficients  $c1$  to  $cN$  are general FIR low pass filter coefficients derived using well known general methods. The resulting coefficients  $c1$  to  $cN$  in the coefficient buffer are used to multiply each coefficient by its corresponding DSD signal bit value over a period of time, as described below, to generate the correct values to reproduce the analog audio signal.

[0017] Each bit in the  $N$ -bit delay line has a corresponding coefficient in the  $N$ -coefficient buffer. Thus, the most significant bit **105** in the  $N$ -bit delay line **110** corresponds to coefficient  $c1$  **125a** in the  $N$ -coefficient buffer **120**. An address generator **110** moves from left to right and sequentially accesses each bit in the  $N$ -bit delay line **110**. For each bit in the  $N$ -bit delay line **110**, the address generator **110** determines the corresponding coefficient in the  $N$ -coefficient buffer **120**. Sign controller **130** receives the bit value of the  $N$ -bit delay line **110** currently accessed by the address generator **110** and sets output line **133** high or low depending on whether the bit value is one or zero.

[0018] Multiplier MPY **135** receives at a first input **133** a high or low signal from sign controller **130** indicating the current bit value from the  $N$ -bit delay line **110**. As shown in Figure 1, multiplier MPY **135** receives at a second input **138**, a 32 bit coefficient **125** from  $N$ -coefficient buffer **120**. If the multiplier MPY receives a high signal from sign controller **130**, the multiplier MPY generates a 32 bit binary value representing decimal +1.0 and multiplies this 32 bit binary value with the 32 bit coefficient **125** received at its second input **138**. If the multiplier MPY receives a low signal from sign controller **130**, the multiplier MPY generates a 32 bit binary value representing decimal -1.0 in two's complement form and multiplies this 32 bit binary value with the 32 bit coefficient **125** received at its second input **138**.

[0019] Alternatively, the sign controller **130** may connect to the multiplier MPY **135** through a 32 bit bus (not shown). The sign controller **130** may provide to the multiplier MPY **135** a binary 32 bit value of decimal +1.0 or -1.0 depending on whether the bit value from  $N$ -bit

delay line **110** is one or zero, respectively. In this system, multiplier MPY **135** performs a 32 bit by 32 bit multiply with the output from sign controller **130** and coefficient **125** from *N*-coefficient buffer **120**. The multiplier MPY **135** sends the result of the multiplication via 32 bit output bus **140** to adder **150**.

5    **[0020]**       Accumulator **170** couples to adder **150** through 32 bit bus **160** and initially contains a zero value. The current 32 bit value contained in the accumulator **170** is fed back to the adder **150** to be summed with the next output from multiplier MPY **135**. The result of this addition is then loaded into accumulator **170** and then summed with the next output from multiplier MPY **135**. Once all bits have been processed in the *N*-bit delay line **110** (i.e. the  
10   address generator has reached the least significant bit **185** in *N*-bit delay line **110**) the accumulator **170** contains a 32 bit PCM signal sample that it transmits through output bus **175**. The *N*-bit delay line **110** may then be loaded with the next *N* bit samples from the DSD signal for conversion to a PCM signal sample.

**[0021]**       As described above, the DSD to PCM conversion device **100** requires *N*  
15   multiplies to convert *N* DSD bit samples to a multiple bit PCM signal sample. Multiplier MPY **135** may be an odd sized multiplier with one 32 bit input bus and a single bit input line. Because a single bit is processed from the *N*-bit delay line **110** through the multiplier MPY **135** and accumulator **170** at a time, multiple cycles may be required to generate the multiple bit PCM signal sample **180**.

20   **[0022]**       Figure 2 shows another implementation, in accordance with the preferred embodiment of the invention, of a DSD to PCM conversion device contained in a digital signal processor (DSP) **200**. This device uses a look-up table **260** that, preferably, may be stored on a memory **210**. The memory may be a Read-Only-Memory (ROM) that stores a permanent copy of the look-up table or a Random Access Memory (RAM) in which the look-up table may be built,  
25   as described below, each time power is applied to the DSP. In some preferred embodiments of the invention, the memory is located in the DSP **200**. In alternative embodiments of the invention, the memory **210** is an external memory that couples to the DSP **200** through a multiple bit bus.

**[0023]**       The DSD to PCM conversion device shown in Figure 2 includes a First-In-First-  
30   Out (FIFO) buffer **205** that contains multiple lines each of size *N* bits. Line **202** at the top of the

FIFO buffer contains  $N$  DSD signal sample bits that are subdivided into a number of words Word(0) **212**, Word(1), Word(2), ..., Word( $N/n - 1$ ) **214**. Each word contains  $n$  bits **218** such that line **202** of the FIFO buffer **205** is subdivided into  $N/n$  words. Thus, for example, if line **202** of the FIFO buffer **205** contains  $N=32$  bits, and as shown in Figure 2 each word contains  $n=8$  bits, there would be 4 words ( $N/n=32/8=4$ ) in line **202**.

[0024] Look-up table **210** may be organized as a two dimensional array data structure containing section 0 **262**, section 1, section 2, ... , section( $N/n - 1$ ) **265**. Each section may include a  $2^n$  entry array **270** containing sum[0] **275**, sum[1], sum[2], ..., sum[ $2^n-1$ ] **280**. As shown in Figure 2, each entry of the array sum **270** may contain a 32 bit binary value **279**. As described in greater detail below and shown in Figure 3, each 32 bit binary value in the array **270** is a precomputed partial sum for one of the bit patterns index=00...00b **276a**, 00...01b **276b**, 00...10b **276c**, ..., 11...11b **276n** using the corresponding coefficient for each bit in the bit pattern to calculate the sum. The look-up of a word Word(0) ... Word( $N/n - 1$ ) in the table **260** includes determining the section corresponding to the word in the FIFO buffer **205** and matching the bit pattern of the word to a bit pattern in the array **270** to determine the precomputed partial sum. Thus, for the example described above where  $N=32$  bits and  $n=8$  bits, the look-up table **210** would contain four sections section 0, section 1, section 2, and section 3. Each section would include 256 entries ( $2^n=2^8=256$  entries) in array **270** starting at sum[0], sum[1], sum[2], ..., sum[255]. Each entry of the array sum would contain a 32 bit precomputed sum for one of the bit patterns index=00...00b **276a**, 00...01b **276b**, 00...10b **276c**, ..., 11...11b **276n**. Thus, if the  $n$ -bit word from the FIFO buffer **205** was Word(0)=11001101b **212**, section 0 corresponding to Word(0) would be accessed (see below for description) and a look-up of the appropriate entry in array **270** would be performed. Look-up of the appropriate entry in array **270** includes matching Word(0)=11001101b **212** to one of the bit patterns 00000000b , 00000001b, 00000010b, ..., 11111111b to determine the 32 bit precomputed sum. The index of each entry in the array sum **270** sum[index=0], sum[index=1], ..., sum[index= $2^n-1$ ] corresponds to the bit pattern for each entry and so Word(0)=11001101b=205dec would retrieve the 32 bit precomputed sum at sum[205].

[0025] As mentioned above, each word in the FIFO buffer has a corresponding section in the look-up table **260**. Thus, the most significant Word(0) **212** in the FIFO buffer **205**



corresponds to section 0 **262** in the look-up table **260**. An address generator **220** moves from left to right and sequentially accesses each word in the FIFO buffer **205**. For each word in the FIFO buffer **205**, the address generator **220** determines the corresponding section in the look-up table **260**. The bit pattern of the n-bit word accessed by the address generator is also passed to the  
 5 corresponding section through bus **230** so that a look-up in array **270** for the precomputed sum can be performed.

**[0026]** The resulting 32 bit precomputed sum from the look-up table **260** for the corresponding word from FIFO buffer **205** is provided to output bus **235** and added in adder **240** to the current value of accumulator **255**. Preferably, accumulator **255** is initially set to a zero  
 10 value. The current 32 bit value contained in the accumulator **255** is fed back to the adder **240** through bus **245** to be summed with the next 32 bit precomputed sum from look-up table **260**. The result of this addition is then loaded into accumulator **255** through bus **250** and then summed with the next output from look-up table **260**. Once all words have been processed in the current line **202** of the FIFO buffer **205** (i.e. the address generator has reached the least significant word  
 15 **214** in FIFO buffer **205**) the accumulator **255** contains a 32 bit PCM signal sample that it transmits through output bus **285**. The FIFO buffer **205** then flushes line **202** from the FIFO buffer and moves the next line below line **202** to the top of the FIFO buffer for conversion from DSD signal bits to a PCM signal sample.

**[0027]** Turning now to Figure 3, a flow chart **300** for generating the look-up table shown  
 20 in Figure 2 in accordance with the preferred embodiment of the invention is shown. Generating the look-up table **260** includes determining the precomputed sums for each of the  $2^n$  bit patterns index=00..00b ... 11..11b in each section of the look-up table **260**. As mentioned above, for each word in the FIFO buffer **205**, the address generator determine the appropriate section and the bit pattern of the n-bit word is matched with one of the  $2^n$  bit patterns in array **270** of the  
 25 section.

**[0028]** Each bit position in the n-bit words of the FIFO buffer has a corresponding coefficient in a two dimensional coefficient array of size  $\text{coeff}[N/n][n]$ . Thus, a total of N coefficients exist for each of the N bits in the FIFO buffer **205**. The coefficients in the coefficient array are accessed based on section for N/n sections and bit position i within the n bits  
 30 of a word such that  $\text{coeff}[\text{section}][i]$  corresponds to a bit position i of word(section). Thus, the

most significant bit  $i=0$  in the most significant Word(section=0) 212 in the FIFO buffer 205 corresponds to  $\text{coeff}[\text{section}=0][i=0]$  in the coefficient array. As mentioned above, each coefficient  $\text{coeff}[\text{section}][i]$  in the coefficient array is a constant that is a general FIR low pass filter coefficient derived using well known general methods.

5    **[0029]**       Referring to Figure 3, generating the look-up table is performed using the following technique. A variable section corresponding to the sections shown in look-up table 260 of Figure 2 is initialized to zero at block 315. Similarly, variable index is initialized to zero in block 320. Variable index corresponds to the  $n$ -bit index of the two dimensional array table for look-up table 260  $\text{table}[\text{section}=0][\text{index}=0]$ ,  $\text{table}[\text{section}=0][\text{index}=1]$ ,  
 10  $\text{table}[\text{section}=0][\text{index}=2]$ , ...  $\text{table}[\text{section}=0][\text{index}=2^n-1]$ . Variable index is also shown in Figure 2 as binary bits  $\text{index}=00...00b$  276a,  $\text{index}=00..01b$  276b,  $\text{index}=00..10b$  276c ...  $\text{index}=11..11b$  276n. In block 330, the variable  $i$  that goes from  $i=0$  to  $i=n$  and keeps track of the current bit in the  $n$ -bit index is also initialized to zero and sum containing the value of the precomputed sum is initialized to zero.

15    **[0030]**       Starting at  $\text{index}=0$  and for each of the  $2^n$  values of index, then for each index starting with the most significant  $i=0$  bit in the index and going to the least significant  $i=n$  bit in the index, the  $2^n$  precomputed sums are determined for each section using the two dimensional coefficient array  $\text{coeff}[\text{section}][i]$ . Thus, in branch condition 335 if the  $i$ 'th bit of index is 1 then go to block 345. In block 345, the  $\text{coeff}[\text{section}][i]$  from the coefficient array is added to the  
 20 current value in sum and  $i$  incremented to the next bit position in index. If the  $i$ 'th bit of index is zero, then branch condition 335 goes to block 340. In block 340, the  $\text{coeff}[\text{section}][i]$  from the coefficient array is subtracted from the current value in sum and  $i$  incremented to the next bit position in index. After all bits in index have been evaluated and the condition  $i=n$  in block 350, the precomputed sum for that index value is present in sum. In block 360, the precomputed sum  
 25 stored in the sum variable is placed into look-up table entry corresponding to  $\text{table}[\text{section}][\text{index}]$  and index incremented to the next bit pattern 276. If the precomputed sums for all values of index from  $\text{index}=0$  to  $\text{index}=2^n$  for the section have been determined and  $\text{index}=2^n$ , the condition in block 370 is not true and the variable section is incremented in block 375. Thus, the precomputed sums for each index value going from  $\text{index}=0$  to  $\text{index}=2^n$  for the  
 30 next section are determined as given above until the precomputed sums for all index values in all

sections of the look-up table have been determined and  $\text{section} = N/n$  in block **380**. Generation of the look-up table **360** is complete and the technique stops in block **390**.

**[0031]** DSP **200** is now ready to receive DSD signal samples for conversion to multiple bit PCM signal samples using the generated look-up table as described with reference to flow chart **400** in Figure 4. Referring also to Figure 2, the FIFO buffer **205** receives  $N$  bits from the DSD signal, and flushes out the oldest  $N$  bits that have been converted to a multiple bit PCM signal sample as shown in block **420**. The  $N$  bits are subdivided into words corresponding to sections as shown in Figure 2. The address generator **220** starting at the most significant word **212** and traversing sequentially from left to right to access each word, determines the correct precomputed sum by performing a look-up in table **260** to match the bit pattern in word to the correct bit pattern  $00..00b$  **276a** ...  $11..11b$  **276n**. Thus, in block **440**, a look-up of the two dimensional array  $\text{table}[\text{section}][\text{word}[\text{section}]]$  containing the precomputed sums is performed and added to the sum. In the two dimensional table array,  $\text{word}[\text{section}]$  corresponds to the bit pattern in the word. Thus, as shown in Figure 2 for  $\text{section}=0$ ,  $\text{word}[\text{section}=0]=11001101b=205_{\text{dec}}$  and element  $\text{table}[\text{section}=0][\text{word}[\text{section}=0]=205]$  contains the precomputed sum corresponding to  $\text{word}[0]$ .

**[0032]** After adding the precomputed sum for the word to sum, the variable section is incremented to determine the precomputed sum for the next word and this value is added to the sum variable. Thus, in block **450** after all words have been evaluated and the address generator has reached the  $\text{section}(N/n - 1)$  in the look up table **260** and corresponding  $\text{word}(N/n - 1)$  **214**, the condition is not true since  $\text{section} = N/n$ . Block **460** is evaluated and the multiple bit PCM signal sample in variable sum is generated. Finally, in block **470** if input continues from the DSD signal, then  $N$  DSD samples are loaded into the FIFO buffer as given in block **420** and the oldest  $N$  bits are flushed out. If no more bits from the DSD signal are in the FIFO buffer for conversion to PCM signal samples, the conversion technique stops in block **480**.

**[0033]** The technique described above for conversion of DSD signals to PCM signals reduces the sampling rate for the PCM signal samples. Thus, if the DSD signal is sampled at a rate of 2.8224 MHz and  $N$  DSD samples are converted to one PCM signal sample, the PCM signal sampling rate is decimated to  $2.8224/N$  MHz. Common values of  $N=16, 32$ , and  $64$  would give respective sampling rates of 176.4 KHz, 88.2 KHz, and 44.1 KHz.

**[0034]** While the present invention has been described with respect to a limited number of embodiments, those skilled in the art will appreciate numerous modifications and variations therefrom. Thus, in an alternative embodiment, the two dimensional array table may be a partial linked list data structure with each section containing a one dimensional array of precomputed sums and pointing to the address of the next sequential section in memory. Use of the linked list data structure allows use of non-contiguous blocks of memory in the DSP. It is intended that the appended claims cover all such modifications and variations as fall within the true spirit and scope of this present invention.